

CLAIMS

1. (Previously presented) A method comprising:

initializing a computer system, the computer system including a first processor and a second processor;

designating the first processor to handle a polling function for a timer interrupt process for the computer system, a normal execution thread to be processed by the second processor;

setting a timer for a plurality of time intervals for the timer interrupt process;

calling a polling function at the end of each of the plurality of time intervals, the polling function being performed by the first processor, the polling function to determine if any special events have occurred; and

if the polling function results in a positive result, processing the results of the polling function with the second processor.
2. (Previously presented) The method of claim 1, wherein the polling function comprises event handling for a network stack by polling a network interface card (NIC) of the computer system.
3. (Previously presented) The method of claim 1, wherein the first processor is an application processor for the computer system.
4. (Original) The method of claim 3, further comprising declaring the first processor to be dedicated to the polling function.

5. (Previously presented) The method of claim 1, wherein the second processor is a bootstrap processor for the computer system.
6. (Previously presented) The method of claim 1, wherein the normal execution thread is processed by the second processor in parallel at least in part with performance of the polling function by the first processor.
7. (Previously presented) The method of claim 1, wherein the timer interrupt process is the only method of asynchronous event handling for the first processor and the second processor in the computer system.
8. (Previously presented) An event handling mechanism for a computer system comprising:
 - a first processor, the first processor designated to handle a timer interrupt process, the first processor to perform a polling operation for event handling each time an interrupt timer reaches a specified time interval; and
 - a second processor, the second processor to perform a normal processing operation, the first processor to transfer data to the second processor for processing if the polling operation provides a positive result.
9. (Previously presented) The event handling mechanism of claim 8, wherein the performance of the polling operation by the first processor overlaps at least in part with the performance of the normal processing operation by the second processor.
10. (Original) The event handling mechanism of claim 8, wherein the first processor is dedicated to event handling.

11. (Previously presented) The event handling mechanism of claim 8, wherein an event for the polling operation for the computer system comprises a network stack event and wherein the first processor polls a network interface card (NIC) of the computer system.

12. (Previously presented) The event handling mechanism of claim 8, wherein the first processor and the second processor are separate physical processors of the computer system.

13. (Previously presented) The event handling mechanism of claim 8, wherein the first processor and the second processor are logical processors in a single physical processor of the computer system.

14. (Previously presented) A computer system comprising:

a first processor, the first processor being designated to perform an event handling function for the computer system;

a second processor, the second processor to perform a processing function for the computer system;

a timer, the timer being set for a time interval for the event handling function of the first processor, a function call for the first processor being called at the end of the time interval for polling of events; and

a memory, the first processor writing data relating to events to the memory to transfer the data to the second processor for processing.

15. (Original) The computer system of claim 14, wherein the second processor is a bootstrap processor.

16. (Original) The computer system of claim 14, wherein the first processor is an application processor.
17. (Original) The computer system of claim 14, wherein the first processor and the second processor operate in parallel at least in part.
18. (Previously presented) The computer system of claim 14, wherein the timer provides the only event handling mechanism for the computer system.
19. (Original) The computer system of claim 14, wherein the computer system comprises a single-threaded processing environment.
20. (Previously presented) The computer system of claim 14, wherein the computer system is a multi-processor system, and wherein the first processor is a first physical processor and the second processor is a second physical processor.
21. (Previously presented) The computer system of claim 14, wherein the computer system is a hyper-threaded system, and wherein the first processor is a first logical processor of a physical processor and the second processor is a second logical processor of the physical processor.
22. (Previously presented) A machine-readable medium having stored thereon data representing sequences of instructions that, when executed by a processor, cause the processor to perform operations comprising:
- initializing a computer system, the computer system including a first processor and a second processor;

designating the first processor to handle a polling function for a timer interrupt process for the computer system, a normal execution thread to be processed by the second processor;

setting a timer for a plurality of time intervals for the timer interrupt process;

calling a polling function at the end of each of the plurality of time intervals, the polling function being directed to the first processor, the polling function to determine if any special events have occurred; and

if the polling function results in a positive result, directing the results of the polling function to the second processor.

23. (Previously presented) The medium of claim 22, wherein the polling function comprises polling a computer interface of the computer system.

24. (Previously presented) The medium of claim 22, wherein the first processor is an application processor for the computer system.

25. (Original) The medium of claim 22, wherein the instructions further comprise instructions that, when executed by a processor, cause the processor to perform operations comprising:

declaring the first processor to be dedicated to the polling function.

26. (Previously presented) The medium of claim 22, wherein the second processor is a bootstrap processor for the computer system.

27. (Previously presented) The medium of claim 22, wherein processing of a normal execution thread by the second processor overlaps in time at least in part with performance of the polling function by the first processor.